

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号
特開2001-125610
(P2001-125610A)

(43)公開日 平成13年5月11日(2001.5.11)

| (51)Int.Cl. ⁷ | 識別記号 | F I | テームコード*(参考) |
|--------------------------|-------|---------------|-------------|
| G 0 5 B | 19/05 | G 0 5 B 19/02 | W 5 H 2 1 5 |
| | 15/02 | 19/05 | G 5 H 2 1 9 |
| | 19/02 | 15/02 | P 5 H 2 2 0 |
| | | 19/05 | A 9 A 0 0 1 |

審査請求 未請求 請求項の数5 O L (全 12 頁)

(21)出願番号 特願平11-303578

(22)出願日 平成11年10月26日(1999.10.26)

(71)出願人 000002945

オムロン株式会社

京都市下京区塩小路通堀川東入南不動堂町
801番地

(72)発明者 市村 勝彦

京都府京都市右京区花園土堂町10番地 オ
ムロン株式会社内

(74)代理人 100069431

弁理士 和田 成則

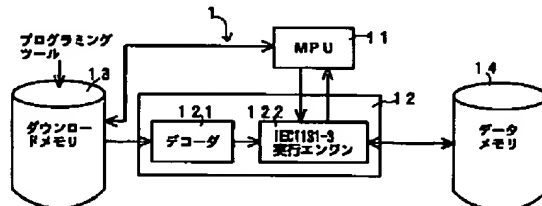
最終頁に続く

(54)【発明の名称】 制御装置、およびこの制御装置が実行するためのプログラムを生成するプログラミングツール

(57)【要約】

【課題】 タスクで構成されたユーザプログラムの実行処理においてオーバヘッド時間を少なくして性能を向上させる制御装置を提供する。

【解決手段】 MPU 11は、INリフレッシュ処理を行うと、その後、アクセラレータ12に対して起動を駆ける。アクセラレータ12は、データメモリ14に有するワークエリアに設定されたタスク番号“i”に該当する指定タスクを実行するか否かを判断する。アクセラレータ12は、タスク番号“i”に該当する指定タスクが非実行のタスクであると判断した場合には、このタスクに実行を行わない。一方、アクセラレータ12は、タスク番号“i”に該当する指定タスク51が非実行のタスクであると判断した場合には、ダウンロードメモリ13から#iのタスク51を読み出し、これを実行する。



【特許請求の範囲】

【請求項1】 MPUに代替して所定の処理を実行するアクセラレータを備えた制御装置において、

IEC1131-3で準拠したプログラム言語により記述されたタスクを備えるユーザプログラムを記憶するユーザプログラム記憶手段と、

IEC1131-3で準拠したプログラム言語により記述された前記タスクの実行条件を記述するシステムタスクを備えるシステムタスクプログラムを記憶するシステムタスクプログラム記憶手段とを具備しており、

前記アクセラレータは、

前記システムタスクプログラム記憶手段から読み出されたシステムタスクに基づき、タスクを実行するか否かを判断するタスク実行判断手段と、

このタスク実行判断手段によりタスクが実行されると判断された場合には、前記タスク記憶手段からタスクを読み出して実行するタスク実行手段とを有することを特徴とする制御装置。

【請求項2】 MPUに代替して所定の処理を実行するアクセラレータを備えた制御装置において、

IEC1131-3で準拠したプログラム言語により記述されたタスクを備えるユーザプログラムを記憶するユーザプログラム記憶手段と、

ラダープログラム、C言語等の専用言語で記述された前記タスクの実行条件を記述するシステムタスクを備えるシステムタスクプログラムを記憶するシステムタスクプログラム記憶手段とを具備しており、

前記アクセラレータは、

前記システムタスクプログラム記憶手段から読み出されたシステムタスクに基づき、タスクを実行するか否かを判断するタスク実行判断手段と、

このタスク実行判断手段によりタスクが実行されると判断された場合には、前記タスク記憶手段からタスクを読み出して実行するタスク実行手段とを有することを特徴とする制御装置。

【請求項3】 前記タスク実行判断手段は、

サイクル数カウンタのカウンタ値からタスクの開始サイクルを減算した値をタスク周期サイクルで除算し、その剰余が“0”である場合にはタスクを実行し、そうでない場合には実行しないと判断することを特徴とする請求項1または2記載の制御装置。

【請求項4】 制御装置が実行するプログラムを生成するプログラミングツールにおいて、

IEC1131-3で準拠したプログラム言語により記述されたユーザプログラムをコンパイルしてタスクを備える前記制御装置が実行可能なユーザプログラムを生成するユーザプログラム生成手段と、

IEC1131-3で準拠したプログラム言語により記述されたシステムタスクプログラムをコンパイルしてシステムタスクを備える前記制御装置が実行可能なシステ

ムタスクプログラムを生成するシステムタスクプログラム生成手段とを具備することを特徴とするプログラミングツール。

【請求項5】 制御装置が実行するプログラムを生成するプログラミングツールにおいて、

IEC1131-3で準拠したプログラム言語により記述されたユーザプログラムをコンパイルしてタスクを備える前記制御装置が実行可能なユーザプログラムを生成するユーザプログラム生成手段と、

ラダープログラム、C言語等の専用言語で記述されたシステムタスクプログラムをコンパイルしてシステムタスクを備える前記制御装置が実行可能なシステムタスクプログラムを生成するシステムタスクプログラム生成手段とを具備することを特徴とするプログラミングツール。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、MPUに代替して所定の処理を高速に実行するアクセラレータを備えた制御装置およびこの制御装置が実行するためのプログラムを生成するプログラミングツールに関する。

【0002】

【従来の技術】制御装置としてのプログラマブルロジックコントローラ（以下、PLCという）は、ユーザプログラムとしてラダープログラムが使用されてきたが、近年、IEC1131-3に準拠したユーザプログラムが使用されるとともに、MPUに代替して所定の処理を高速に実行するアクセラレータを備えた構成のものが稼働するようになってきた。

【0003】ユーザプログラム50は、図8に示すように、IEC1131-3で準拠したプログラム言語により記述された#1～#nのタスク51で構成されており、タスク制御プログラム60の制御に基づき、#1～#nのタスク51が実行されるようになっている。

【0004】アクセラレータを備えた従来の制御装置としてのPLC1は、図9に示すように、MPU11と、アクセラレータ12と、ダウンロードメモリ13と、データメモリ14とを備えて構成されている。

【0005】MPU11は、この実施形態のPLC1全体を制御するようになっている。

【0006】アクセラレータ12は、MPU11と代替して所定の処理を高速に実行するものであって、ダウンロードメモリ13に記憶されているユーザプログラム50から読み出されたタスク51を解読するデコーダ121と、デコーダ121で解読したタスク51を実行して、必要がある場合にはデータメモリ14に対してデータを送受信するIEC1131-3実行エンジン122とを有して構成されている。

【0007】ダウンロードメモリ13は、図10に示すように、後述するプログラミングツールからタスク制御プログラム60、タスク設定データおよびユーザプロ

ラム50のそれぞれがダウンロードされたタスク制御プログラムファイル131、タスク設定ファイル132およびユーザプログラム50を有するユーザプログラムファイル133とを備えている。

【0008】タスク制御プログラムファイル131は、オブジェクトモジュールからなるタスク制御プログラム60を有するものであり、タスク設定ファイル132は、タスク制御プログラム60が実行するタスク51を指定するタスク設定データを有するものである。

【0009】タスク設定データには、図11に示すように、ユーザプログラム50を構成するタスク51を指定するタスク番号、各タスクの優先度、各タスク51がいつ開始するかを規定する開始サイクル、および何サイクルごとにタスク51を実行するかを規定する情報を有している。

【0010】図11に、A欄で示したタスク番号が“1”、優先度が“1”、開始サイクルが“0”、実行周期が“3”のタスク51は、図12に示すように、実行開始時期が0サイクルの時点であって、3サイクルごとに繰り返し実行することを示している。

【0011】B欄で示したタスク番号が“2”、優先度が“1”、開始サイクルが“0”、実行周期が“5”のタスク51は、実行開始時期が0サイクルの時点であって、5サイクルごとに繰り返し実行することを示している。

【0012】C欄で示したタスク番号が“3”、優先度が“2”、開始サイクルが“1”、実行周期が“6”のタスク51は、実行開始時期が1サイクルの時点であって、6サイクルごとに繰り返し実行することを示している。

【0013】ユーザプログラムファイル133は、#1～#nのタスク51で構成されているユーザプログラム50を有しているものである。

【0014】このPLC1の動作について、図13を参照して説明する。

【0015】MPU11は、プログラミングツールにより生成されたタスク制御プログラムファイルのタスク制御プログラム60、タスク設定ファイルのタスク設定データ、およびユーザプログラムファイルのIEC1131-3で準拠された#1～#nのタスク51を有するユーザプログラム50のそれぞれを、ダウンロードメモリ13に有するタスク制御プログラムファイル131、タスク設定ファイル132およびユーザプログラムファイル133にダウンロードする(ステップ310)。

【0016】次に、MPU11は、I/O割り付け等の初期設定を行い(ステップ315)、PLC1が起動されているか否かを判断する(ステップ320)。

【0017】MPU11は、起動されていないと判断した場合には(ステップ320; No)、実行を停止して(ステップ375)処理を終了し、一方、起動されてい

ると判断した場合には(ステップ330; Yes)、I/Nリフレッシュ処理を行い(ステップ325)、タスク番号を初期化する(ステップ330)。

【0018】次に、MPU11は、ダウンロードメモリ13に記憶されているタスク設定ファイル132に有するタスク設定データに基づき、該当するタスク51を実行するか否かを判断し(ステップ335)、その結果、タスク51を実行しないと判断した場合には(ステップ335; No)、ステップ360に移行し、後述する以下のような処理を行う。

【0019】一方、MPU11は、タスク51を実行すると判断した場合には(ステップ335; Yes)、実行するタスク番号をアクセラレータ12に通知する(ステップ340)。

【0020】アクセラレータ12は、MPU11からタスク番号の通知を受けると起動を開始し(ステップ345)、通知されたタスク番号のタスク51をダウンロードメモリ13中に有するユーザプログラムファイル133から読み出して実行する(ステップ350)。

【0021】その後、MPU11は、アクセラレータ12に対して通知したタスク51の実行が終了したか否かを、アクセラレータ12からの終了メッセージを受けたか否かで判断する(ステップ355)。

【0022】MPU11は、アクセラレータ12に対して通知したタスク51の実行が未だ終了していないと判断した場合には(ステップ355; No)、アクセラレータ12からの終了メッセージを受けるまで待機する。

【0023】一方、MPU11は、アクセラレータ12に対して通知したタスク51の実行が終了したと判断した場合には(ステップ355; Yes)、さらに、ユーザプログラム50による実行処理が終了したか否かを判断する(ステップ360)。

【0024】MPU11は、ユーザプログラム50による実行処理が終了していないと判断した場合には(ステップ360; No)、タスク番号をインクリメントし(タスク番号に1を加算する)(ステップ365)、ステップ335に移行して上述した同様な処理を行う。

【0025】一方、MPU11は、ユーザプログラム50による実行処理が終了したと判断した場合には(ステップ360; Yes)、O/Nリフレッシュ処理を行い(ステップ370)、ステップ320に移行し上述したと同様な処理を行う。

【0026】次に、上述したPLC1が実行可能なタスク制御プログラム60、タスク設定データ、およびユーザプログラム50を生成する従来のプログラミングツールについて説明する。

【0027】このプログラミングツール2は、図14に示すように、表示部21と、入力部22と、標準入出力I/F23と、タスク制御プログラム生成手段24と、タスク設定データ生成手段25と、タスク生成手段26

と、メモリ27と、出力I/F28とを備えて構成されている。

【0028】表示部21は入力内容および処理内容を表示するようになっており、入力部22は、ソースプログラムとしてのタスク制御プログラム、タスク設定データ、およびソースプログラムとしてのユーザプログラムを入力するようになっている。

【0029】標準出力I/F23は、表示部21および入力部22と、タスク制御プログラム生成手段24、タスク設定データ生成手段25およびタスク生成手段26との間をインターフェースするようになっている。

【0030】タスク制御プログラム生成手段24は、入力部22から入力されたソースプログラムとしてのタスク制御プログラムをコンパイルし、オブジェクトモジュールのタスク制御プログラムを生成するようになっている。

【0031】そして、タスク制御プログラム生成手段24は、生成したタスク制御プログラム60をメモリ27に有するタスク制御プログラムファイル271に格納するようになっている。

【0032】タスク設定データ生成手段25は、入力部22から入力されたタスク設定データを基にして、制御装置が実行可能なタスク設定データを生成するようになっている。

【0033】そして、タスク設定データ生成手段25は、生成したタスク設定データをメモリ27に有するタスク設定ファイル272に格納するようになっている。

【0034】タスク生成手段26は、入力部22から入力されたソースプログラムとしてのタスクをコンパイルし、オブジェクトモジュールのタスク51を生成するようになっている。

【0035】そして、このタスク生成手段26は、生成したタスク51をメモリ27に有するユーザプログラムファイル273に格納するようになっている。

【0036】メモリ27は、上述したように、タスク制御ファイル271、タスク設定ファイル272、およびユーザプログラムファイル273を記憶するものである。

【0037】出力I/F28は、メモリ27に記憶されているタスク制御プログラムファイル271、タスク設定ファイル272およびユーザプログラムメモリ273のそれぞれに記憶されているタスク制御プログラム60、タスク設定データおよびユーザプログラム50をPLC1を構成するダウンロードメモリ13に出力するインターフェースである。

【0038】このプログラミングツール2の処理動作を、図15を参照して説明する。

【0039】このプログラミングツール2は、入力部22よりソースプログラムからなるタスク制御プログラムが入力されると(ステップ410)、入力されたソース

プログラムのタスク制御プログラムをコンパイルし(ステップ415)、オブジェクトモジュールのソースプログラムのタスク制御プログラム60生成し(ステップ420)、これをメモリ27に記憶する(ステップ430)。

【0040】次に、プログラミングツール2は、入力部22よりソースプログラムからなるユーザプログラムが入力されると(ステップ440)、入力されたソースプログラムのユーザプログラムをコンパイルし(ステップ445)、オブジェクトモジュールのユーザプログラム50を生成し(ステップ450)、これをメモリ27に記憶する(ステップ460)。

【0041】続いて、プログラミングツール2は、入力部22からタスク設定データが入力されると(ステップ465)、入力されたタスク設定データを用いてPLC1が実行可能なタスク設定データを生成し(ステップ470)、これをメモリ27に記憶する(ステップ480)。

【0042】その後、プログラミングツール2は、メモリ27中のタスク制御プログラム60、ユーザプログラム50およびタスク設定データを、PLC1のダウンロードメモリ13にダウンロードし(ステップ490)、処理を終了する。

【0043】

【発明が解決しようとする課題】アクセラレータを備えた従来の制御装置としてのPLC1では、上述したように、アクセラレータ12が、MPU11から実行すべきタスク番号の通知を受けとった後にそのタスク51を実行し、かつ、その後、そのタスク51の処理を終了すると、終了した旨をMPU11に通知する必要があるもので、ユーザプログラム50が多くのタスク51で構成されている場合には、これら通知処理によるオーバーヘッドの累積により性能低下を招くという問題点があった。

【0044】また、IEC1131-3の内容は、タスク51の記述の仕方が主であり、タスク51の制御に関する内容が厳密に規定されていないので、タスク制御プログラム60は、必ずしもIEC1131-3に準拠されたものでなく、それ以外の専用言語、例えばラダープログラム、C言語等の専用言語で記述したフォーマットものが使用されている。

【0045】従って、タスク制御プログラムがベンダー毎に独自のフォーマットで記述されていたのでは、ベンダーの変更があった場合、新たにベンダーになった人がプログラム変更を行うに際して、プログラム変更をスムーズに行えないおそれがあるという問題点があった。

【0046】そこで、本発明は上述した問題点に鑑み、タスクで構成されたユーザプログラムの実行処理においてオーバーヘッド時間を少なくして性能を向上させる制御装置を提供することを目的とする。

【0047】また、本発明は、ベンダーの変更があった場合、新たにベンダーになった人がプログラム変更を行うに際して、プログラム変更を容易に行うことができる制御装置が実行するためのプログラムを生成するプログラミングツールを提供することを目的とする。

【0048】

【課題を解決するための手段】上述した目的を達成するため、請求項1記載の発明は、MPUに代替して所定の処理を実行するアクセラレータを備えた制御装置において、IEC1131-3で準拠したプログラム言語により記述されたタスクを備えるユーザプログラムを記憶するユーザプログラム記憶手段と、IEC1131-3で準拠したプログラム言語により記述された前記タスクの実行条件を記述するシステムタスクを備えるシステムタスクプログラムを記憶するシステムタスクプログラム記憶手段とを具備しており、前記アクセラレータが、前記システムタスクプログラム記憶手段から読み出されたシステムタスクに基づき、タスクを実行するか否かを判断するタスク実行判断手段と、このタスク実行判断手段によりタスクが実行されると判断された場合には、前記タスク記憶手段からタスクを読み出して実行するタスク実行手段とを有するようにする。

【0049】請求項2記載の発明は、MPUに代替して所定の処理を実行するアクセラレータを備えた制御装置において、IEC1131-3で準拠したプログラム言語により記述されたタスクを備えるユーザプログラムを記憶するユーザプログラム記憶手段と、ラダープログラム、C言語等の専用言語で記述された前記タスクの実行条件を記述するシステムタスクを備えるシステムタスクプログラムを記憶するシステムタスクプログラム記憶手段とを具備しており、前記アクセラレータが、前記システムタスク記憶手段から読み出されたシステムタスクに基づき、タスクを実行するか否かを判断するタスク実行判断手段と、このタスク実行判断手段によりタスクが実行されると判断された場合には、前記タスク記憶手段からタスクを読み出して実行するタスク実行手段とを有するようにする。

【0050】請求項3記載の発明は、請求項1または2記載の発明において、前記タスク実行判断手段が、サイクル数カウンタのカウント値からタスクの開始サイクルを減算した値をタスク周期サイクルで除算し、その剰余が“0”である場合にはタスクを実行し、そうでない場合には実行しないと判断するようにする。

【0051】請求項4記載の発明は、制御装置が実行するプログラムを生成するプログラミングツールにおいて、IEC1131-3で準拠したプログラム言語により記述されたユーザプログラムをコンパイルしてタスクを備える前記制御装置が実行可能なユーザプログラムを生成するユーザプログラム生成手段と、IEC1131-3で準拠したプログラム言語により記述されたシステ

ムタスクプログラムをコンパイルしてシステムタスクを備える前記制御装置が実行可能なシステムタスクプログラムを生成するシステムタスクプログラム生成手段とを具備するようにする。

【0052】請求項5記載の発明は、制御装置が実行するプログラムを生成するプログラミングツールにおいて、IEC1131-3で準拠したプログラム言語により記述されたユーザプログラムをコンパイルしてタスクを備える前記制御装置が実行可能なユーザプログラムを生成するユーザプログラム生成手段と、ラダープログラム、C言語等の専用言語で記述されたシステムタスクプログラムをコンパイルしてシステムタスクを備える前記制御装置が実行可能なシステムタスクプログラムを生成するシステムタスクプログラム生成手段とを具備するようにする。

【0053】本発明の制御装置では、アクセラレータに有するタスク実行判断手段により、読み出されたシステムタスクに基づき、タスクを実行するか否かを判断するので、従来のように、MPUから実行するタスク番号の通知を受けるとともに、実行したタスクの終了をMPUに通知する必要がなくなる。

【0054】本発明のプログラミングツールでは、タスク生成手段によりIEC1131-3で準拠したプログラム言語により記述された複数のタスクで構成されたユーザプログラムをプログラムをコンパイルしてユーザプログラムを生成するとともに、システムタスクプログラム生成手段によりシステムタスクプログラムをコンパイルしてシステムタスクプログラムを生成して、上述した本発明の制御装置が実行するプログラムを形成する。

【0055】

【発明の実施の形態】以下、本発明に係る制御装置、およびこの制御装置が実行するユーザプログラムを生成するプログラミングツールの実施形態を、図面を参照して説明する。

【0056】＜第1実施形態＞図1は本発明に係る第1実施形態の制御装置としてのプログラマブルロジックコントローラの構成を示すブロック図である。

【0057】なお、この実施形態のPLCにおいて、従来の制御装置としてのPLCと同一な構成部分は、従来のPLCの構成部分と同一な参照符号を付ける。

【0058】この実施形態のPLC1は、IEC1131-3で準拠したプログラム言語で記述された#1～#nのタスクを有するユーザプログラム（後述する）、およびIEC1131-3で準拠したプログラム言語で記述されたものであって、タスク51の制御内容を記述したシステムタスクプログラム（後述する）に基づき、所定処理を実行するものである。

【0059】このPLC1は、図1に示すように、MPU11と、アクセラレータ12と、ダウンロードメモリ13と、データメモリ14とを備えて構成されている。

【0060】MPU11は、この実施形態のPLC1全体を制御するようになっている。なお、制御の詳細については、後述するこの実施形態のPLCの動作説明の箇所を参照する。

【0061】アクセラレータ12は、上述したように、MPU11と代替して所定の処理を高速に実行するものであって、デコーダ121と、IEC1131-1実行エンジン122とから構成されている。

【0062】デコーダ121は、ダウンロードメモリ13のシステムタスクプログラムファイルに記憶されているシステムタスクを解釈し、該当するタスク51が実行すべきものであるか否かを解釈するようになっている。

【0063】また、デコーダ121は、システムタスクを解釈した結果、該当するタスクを実行する場合には、該当するタスクをダウンロードメモリ13のユーザプログラムファイルから読みだし、このタスクを解釈し、これらの解釈結果をIEC1131-1実行エンジン122に出力するようになっている。

【0064】IEC1131-1実行エンジン122は、#1デコーダから受けた解釈結果に基づき、所定の処理を実行し、例えばI/Oデータをデータメモリ14に書き込む処理を行うようになっている。

【0065】ダウンロードメモリ13は、図2に示すように、ユーザプログラムファイル133およびシステムタスクプログラムファイル134を有している。

【0066】ユーザプログラムファイル133は、#1～#nタスク51からなるユーザプログラム50を記憶している。

【0067】システムタスクプログラムファイル134は、図3に示すように、システムタスクプログラム134aを有しており、このシステムタスクプログラム134aには、共通処理としてサイクル数のカウンタを更新する処理を記述したシステムタスク34bと、#1～#nのタスク51の実行内容を記述する#1～#nのシステムタスク134cを有している。

【0068】図みに、#1のシステムタスク134cは、デコーダ121が解釈する内容を有するものであって、その内容として、サイクル数のカウンタからタスクの開始サイクルを減算し、その後、その結果をタスク周期サイクルで除算し、その剰余が“0”である場合には実行し、そうでない場合には実行しないということが記述されている。

【0069】データメモリ14は、IEC1131-1実行エンジン122により出力されたデータ、例えばI/Oデータ、上述したサイクル数カウンタからタスクの開始サイクルを減算した値、この減算した値をタスク周期サイクルで除算したときのその剰余を書き込むようになっている。

【0070】この実施形態の制御装置としてのPLC1の動作について、図4を参照して説明する。

【0071】MPU11は、プログラミングツールにより生成されたシステムタスクプログラムファイルのシステムタスクプログラム、およびユーザプログラムファイルのユーザプログラムのそれぞれを、ダウンロードメモリ13に有するシステムタスクプログラムファイル134、およびユーザプログラムファイル133にダウンロードする(ステップ110)。

【0072】次に、MPU11は、I/O割付け等の初期設定を行い(ステップ120)、起動されているか否かを判断する(ステップ130)。

【0073】MPU11は、起動されていないと判断した場合には(ステップ130;No)、実行を停止して(ステップ195)処理を終了し、一方、起動されていると判断した場合には(ステップ130;Yes)、INリフレッシュ処理を行う(ステップ140)。

【0074】MPU11は、INリフレッシュ処理を行うと、その後、アクセラレータ12に対して起動を駆ける(ステップ145)。

【0075】起動を駆けられたアクセラレータ12は、データメモリ14に有するワークエリアに設定したタスク番号を“1”に設定して初期化する(ステップ150)。

【0076】次に、アクセラレータ12のデコーダ121は、データメモリ14に有するワークエリアに設定されたタスク番号“i”に該当する指定タスクを実行するか否かを判断する(ステップ160)。

【0077】ここで、タスク番号“i”に該当するタスク(以下、指定タスクという)51を実行するか否かの判断を、デコーダ121は、次のようにして行う。

【0078】すなわち、デコーダ121は、システムタスクプログラム134aから#iのシステムタスク134cを読み出し、サイクル数カウンタからこのタスクの開始サイクル数を減算し、この減算値をタスク周期サイクルで除算し、その剰余が“0”である場合には指定タスク51を実行し、“0”以外である場合には指定タスク51を非実行であると判断する。

【0079】アクセラレータ12のデコーダ121は、タスク番号“i”に該当する指定タスク51が非実行のタスクであると判断した場合には(ステップ160;No)、ステップ170に移行する。

【0080】一方、デコーダ121は、タスク番号“i”に該当する指定タスク51が実行するタスクであると判断した場合には(ステップ160;Yes)、ダウンロードメモリ13から#iのタスク51を読み出し、この命令を解釈し、この解釈結果をIEC1131-3実行エンジン122に出力する。

【0081】すると、IEC1131-3実行エンジン122は、解釈結果に基づき#iのタスク51を実行する(ステップ165)。

【0082】その後、アクセラレータ12は、ユーザプ

ログラム50がエンドであるか否かを判断し(ステップ170)、ユーザプログラム50がエンドであると判断した場合には(ステップ170; Yes)、OUTリフレッシュ処理を行い(ステップ190)、ステップ130に処理を移行し、上述した同様な処理を行う。

【0083】一方、アクセラレータ12は、ユーザプログラム50がエンドでないと判断した場合には(ステップ180; Yes)、データメモリ14中のワークエリアに設定されたタスク番号“i”に1を加算し、この加算した値“i+1”を新たにワークエリアに設定し(ス

テップ180)、ステップ160に処理を移行して上述したと同様な処理を続行する。

【0084】この実施形態の制御装置としてのPLC1では、アクセラレータ12により、読み出されたシステムタスク134cに基づき、ユーザプログラム50のタスク51を実行するか否かを判断するので、従来のように、MPU11から実行するタスク番号の通知を受けるとともに、実行したタスク51の終了をMPUに通知する必要がなくなる。

【0085】このため、タスクで構成されたユーザプログラム50の実行処理においてオーバーヘッド時間を少なくして性能を向上させることができる。

【0086】<第2実施形態>次に、本発明に係る制御装置の第2実施形態の構成について説明する。

【0087】図5は本発明に係る制御装置としてのPLCの第2実施形態の構成を示すブロック図である。

【0088】なお、この実施形態のPLC1において、上述したPLCと同一な構成部分は、従来のPLCの構成部分と同一な参照符号を付ける。

【0089】この実施形態のPLC1は、IEC1131-3に準拠した#1~#nのタスク51からなるユーザプログラム59、およびタスク51の制御内容をC、ラダープログラム等の専用言語で記述されたシステムタスクプログラム(後述する)に基づき、所定処理を実行するものである。

【0090】このPLC1は、図5に示すように、MPU11と、アクセラレータ12と、ダウンロードメモリ13と、データメモリ14とを備えて構成されている。

【0091】なお、第1実施形態のPLC1と同一な構成部分については、その詳細説明を省略し、異なる構成部分についてのみ説明する。

【0092】この実施形態の制御装置としてのPLC1は、アクセラレータ12において、#1のデコーダ121と、#2のデコーダ121と、IEC1131-3実行エンジン122と、システムタスク実行エンジン123と、アクセラレータ124とを備えて構成されている。

【0093】#1のデコーダ121は、ダウンロードメモリ13中に記憶されているユーザプログラムファイル133から読み出されたタスク51を解説し、この解説

結果をIEC1131-3実行エンジン122に出力するようにになっている。

【0094】#2のデコーダ121は、ダウンロードメモリ13のシステムタスクプログラムファイルに記憶されているシステムタスクを解説し、該当するタスク51が実行すべきものであるか否かを解説するようになっている。

【0095】IEC1131-3実行エンジン122は、#1のデコーダ121から入力されたタスクの解説結果に基づき、タスク51を実行する。

【0096】例えば実行する場合には、IEC1131-3実行エンジン122は、例えばI/Oデータ、計算結果等をマルチプレプレクサ124を介してデータメモリに記憶するようになっている。

【0097】システムタスク実行エンジン123は、#2のデコーダ121から入力されたシステムタスク134cの解説結果に基づき、システムタスク134cの実行、例えば①入力された開始サイクルをサイクルカウンタから減算結果をデータメモリ14にワークエリアに保存したり、②サイクルタイムデータを入力された周期サイクルで除算した結果の剰余をデータメモリ14のワークエリアに保存するようになっている。

【0098】アクセラレータ124は、IEC1131-3実行エンジン122およびシステムタスク実行エンジン123から入力された入力データのうち、どれか1つを選択し、選択した入力データをデータメモリ14に記憶させるようになっている。

【0099】この実施形態のPLC1では、#2のデコーダ121により、C、ラダープログラム等の専用言語で記述されたシステムタスク134cの内容を解説し、システムタスク実行エンジン123が、#2のデコーダ121から入力されたシステムタスク134cの解説結果に基づき、システムタスク134cを実行するので、ベンダが使用したい言語を選択することができ、システムの自由度を高めることができる。

【0100】次に、この制御装置が実行するユーザプログラムを生成するプログラミングツールを、図6を参照して説明する。

【0101】図6は本発明に係るプログラミングツールの一実施形態の構成を示すブロック図である。

【0102】このプログラミングツール2は、図6に示すように、表示部21と、入力部22と、標準入出力I/F23と、タスク生成手段26と、システムタスク生成手段29と、メモリ27と、出力I/F28とを備えて構成されている。

【0103】表示部21は入力内容および処理内容を表示するようになっており、入力部22は、ソースプログラムとしてのタスクおよびソースプログラムとしてのシステムタスクプログラムを入力するようになっている。

【0104】標準入出力I/F23は、表示部21およ

び入力部22と、タスク生成手段26およびシステムタスク生成手段29との間をインターフェースするようになっている。

【0105】タスク生成手段26は、入力部22から入力されたIEC1131-3で準拠したプログラム言語により記述されたソースプログラムからなるタスクをコンパイルし、オブジェクトモジュールのタスク51を生成するようになっている。

【0106】そして、このタスク生成手段26は、生成したタスク51をメモリ27に有するユーザプログラムファイル273に格納するようになっている。

【0107】システムタスク生成手段29は、入力部22から入力されたIEC1131-3で準拠したプログラム言語により記述されたソースプログラムからなるシステムタスクをコンパイルし、オブジェクトモジュールのシステムタスク134b、134cを生成するようになっている。

【0108】そして、このシステムタスク生成手段29は、生成したシステムタスク134b、134cをメモリ27に有するシステムタスクプログラムファイル274に格納するようになっている。

【0109】メモリ27は、上述したように、ユーザプログラムファイル273およびシステムタスクプログラムファイル274を記憶するものである。

【0110】出力I/F28は、メモリ27に記憶されている271、ユーザプログラムファイル273およびシステムタスクプログラムファイル274を、PLC1を構成するダウンロードメモリ13に出力するインターフェースである。

【0111】このプログラミングツール2の処理動作を、図7を参照して説明する。

【0112】このプログラミングツール2は、入力部22よりソースプログラムからなるタスクが入力されると(ステップ210)、入力されたソースプログラムのタスクをコンパイルし(ステップ220)、オブジェクトモジュールのタスク51を生成し(ステップ230)、これをメモリ27のユーザプログラムファイル273に記憶する(ステップ240)。

【0113】次に、プログラミングツール2は、入力部22よりソースプログラムからなるシステムタスクが入力されると(ステップ250)、入力されたソースプログラムのシステムタスクをコンパイルし(ステップ260)、オブジェクトモジュールのシステムタスク134b、134cを生成し、これをメモリ27のシステムタスクプログラムファイル274に記憶する(ステップ250)。

【0114】続いて、プログラミングツール2は、メモリ27中のユーザプログラム50のタスク51、およびシステムタスクプログラムファイル134のシステムタスク134b、134cを、PLC1のダウンロードメ

モリ13にダウンロードし(ステップ280)、処理を終了する。

【0115】システムタスクのプログラムには、上述したように、次の内容のことが記述されている。

【0116】①入力された開始サイクルをサイクルカウンタから減算する内容

②そして、減算した結果をメモリ(図示せず)のワークエリアに保存する内容

③加工済みのサイクルタイムデータを入力された周期サイクルで除算する内容

④除算した結果の剰余をワークエリアに保存する内容

⑤剰余が“0”の場合には加工済みデータを用いて指定タスクを実行させる内容

【0117】この実施形態のプログラミングツール2では、タスク生成手段によりIEC1131-3で準拠したプログラム言語により記述されたソースプログラムからなるタスクをコンパイルしてオブジェクトモジュールのタスクを生成するとともに、システムタスク生成手段によりIEC1131-3で準拠したプログラム言語により記述されたソースプログラムからなるシステムタスクをコンパイルしてシステムタスクを生成するので、上述した本発明の制御装置が実行するプログラムを形成することができる。

【0118】この実施形態のプログラミングツール2は、システムタスク生成手段によりIEC1131-3で準拠したプログラム言語により記述されたソースプログラムからなるシステムタスクをコンパイルしてシステムタスクを生成するようになっているが、これに代えて、システムタスク生成手段によりC、ラダープログラムにより記述されたソースプログラムからなるシステムタスクをコンパイルしてシステムタスクを生成するようにしてもよい。

【0119】

【発明の効果】以上本発明によれば、以下のような効果を有する。

【0120】(1)請求項1記載の発明によれば、アクセラレータに有するタスク実行判断手段により、システムタスクプログラム記憶手段から読み出されたシステムタスクに基づき、タスクを実行するか否かを判断するので、従来のように、MPUから実行するタスク番号の通知を受けるとともに、実行したタスクの終了をMPUに通知する必要がなくなる。

【0121】このため、タスクで構成されたユーザプログラムの実行処理においてオーバーヘッド時間を少なくして性能を向上させることができる。

【0122】(2)請求項2記載の発明によれば、アクセラレータのタスク実行判断手段が、ラダープログラム、C言語等の専用言語で記述されたタスクの実行条件に基づき、タスクを実行するか否かを判断し、そして、タスクを実行してもよいと判断された場合にはタスク実行手

段が、そのタスクを実行するので、ベンダが使用したい言語を選択することができ、システムの自由度を高めることができる。

【0123】(3)請求項3記載の発明によれば、アクセラレータがサイクル数カウンタのカウント値からタスクの開始サイクルを減算した値をタスク周期サイクルで除算し、その剰余が“0”である場合にはタスクを実行し、そうでない場合には実行しないと判断するので、従来のようにMPUから受けるタスク番号の通知待ち時間を省き、処理時間を短縮することができる。

【0124】(4)請求項4記載の発明によれば、タスク生成手段によりIEC1131-3で準拠したプログラム言語により記述されたソースプログラムからなるタスクをコンパイルしてタスクを生成するとともに、システムタスク生成手段によりIEC1131-3で準拠したプログラム言語により記述されたソースプログラムからなるシステムタスクをコンパイルしてシステムタスクを生成するので、上述した請求項1記載の制御装置が実行するプログラムを形成することができる。

【0125】(5)請求項5記載の発明において、ユーザプログラム生成手段によりIEC1131-3で準拠したプログラム言語により記述されたユーザプログラムをコンパイルしてタスクを有するユーザプログラムを生成するとともに、システムタスクプログラム生成手段によりラダープログラム、C言語等の専用言語で記述されたシステムタスクプログラムをコンパイルしてシステムタスクを生成するので、請求項2記載の制御装置が実行するプログラムを形成することができる。

【図面の簡単な説明】

【図1】本発明に係る制御装置の第1実施形態の構成を示すブロック図。

【図2】図1中のダウンロードメモリの構成を示すブロック図。

【図3】図2中のシステムタスクプログラムの構成を示すブロック図。

【図4】この実施形態の制御装置の動作を説明するフローチャート。

【図5】本発明に係る制御装置の第2実施形態の構成を示すブロック図。

【図6】本発明に係るプログラミングツールの一実施形態の構成を示すブロック図。

【図7】この実施形態のプログラミングツールの動作を

説明するフローチャート。

【図8】従来のPLCが実行するユーザプログラムの構成について説明する図。

【図9】従来のPLCに係る一実施形態の構成を示すブロック図。

【図10】ダウンロードファイルの構成を示すブロック図。

【図11】ユーザプログラムを構成するタスク制御プログラムを説明する図。

10 【図12】タスク制御プログラムに基づきタスクを実行したときの例を示す図。

【図13】従来のPLCの処理動作を示すフローチャート。

【図14】従来のプログラミングツールの構成を示すブロック図。

【図15】従来のプログラミングツールの処理動作を示すフローチャート。

【符号の説明】

1 制御装置(PLC)

20 11 MPU

12 アクセラレータ

121 デコーダ

122 IEC1131-3実行エンジン

123 システムタスク実行エンジン

124 マルチプレクサ

13 ダウンロードメモリ

133 ユーザプログラムファイル

134 システムタスクプログラムファイル

14 データメモリ

2 プログラミングツール

21 表示部

22 入力部

23 標準入出力I/F

24 タスク制御プログラム生成手段

25 タスクプログラム生成手段

26 ユーザプログラムメモリ

27 出力I/F

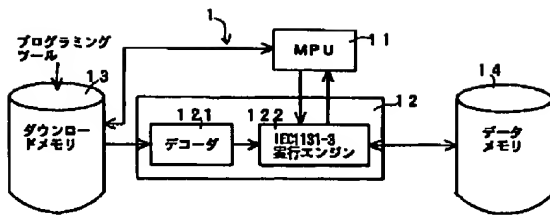
50 ユーザプログラム

51 タスク

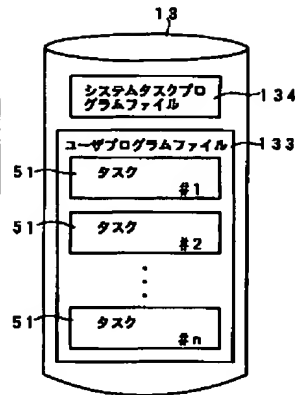
134a システムタスクプログラム

134b, 134c システムタスク

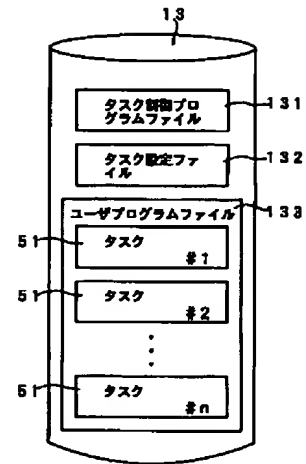
【図1】



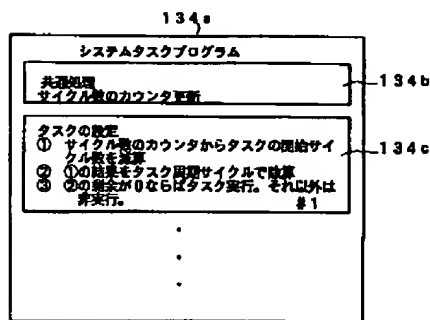
【図2】



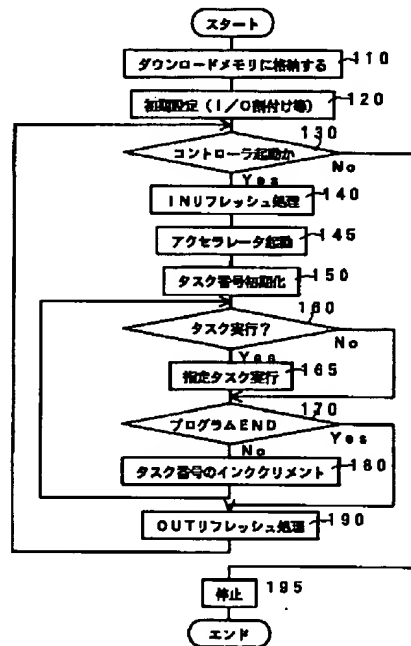
【図10】



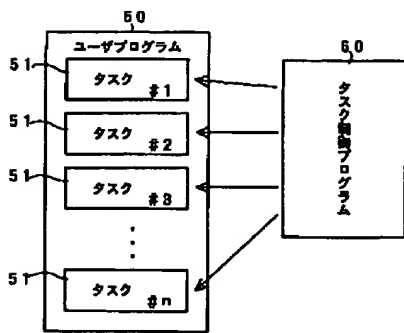
【図3】



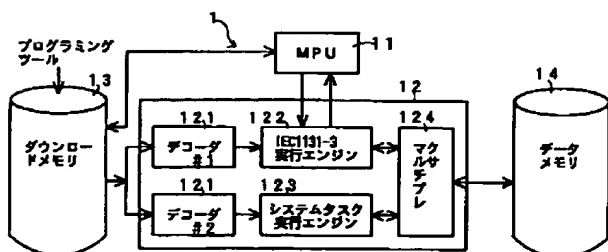
【図4】



【図8】



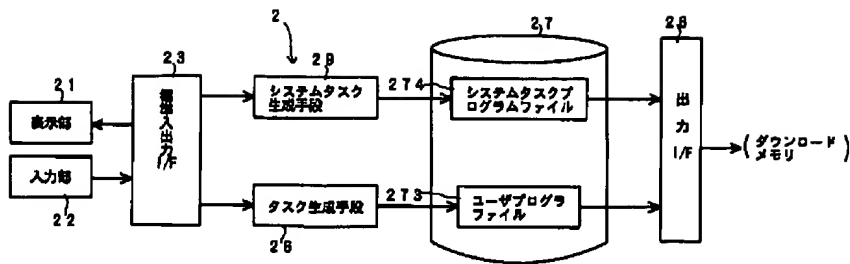
【図5】



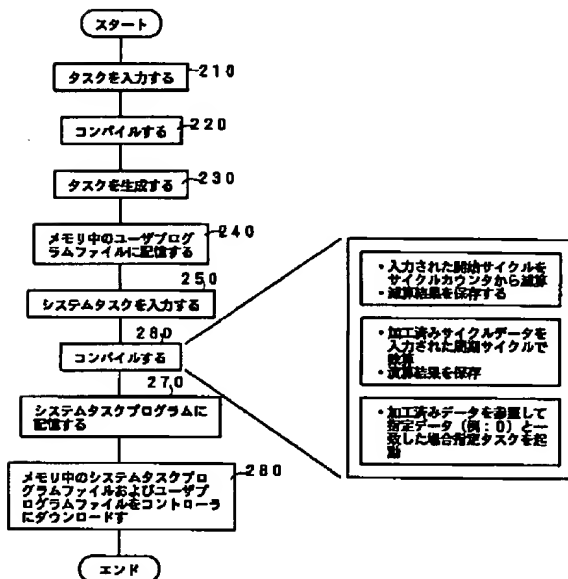
【図11】

| | タスク番号 | 優先度 | 開始サイクル | 実行周期 |
|---|-------|-----|--------|------|
| A | 1 | 1 | 0 | 3 |
| B | 2 | 1 | 0 | 5 |
| C | 3 | 2 | 1 | 8 |
| | ... | ... | ... | ... |

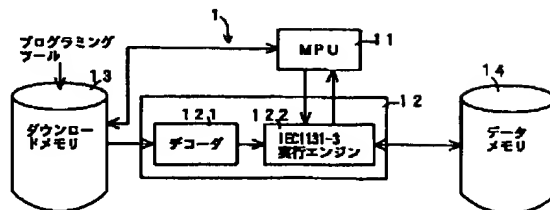
【図6】



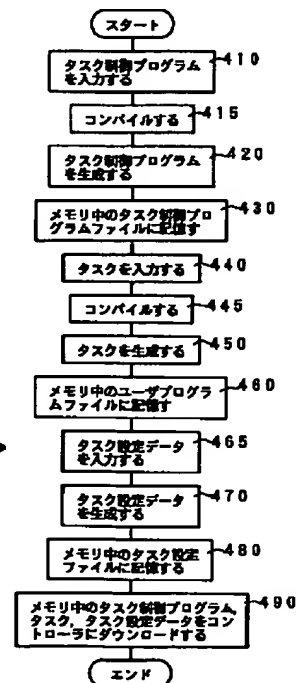
【図7】



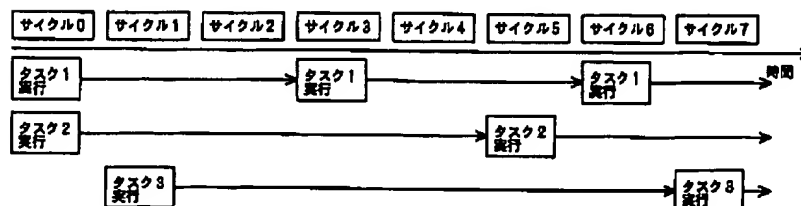
【図9】



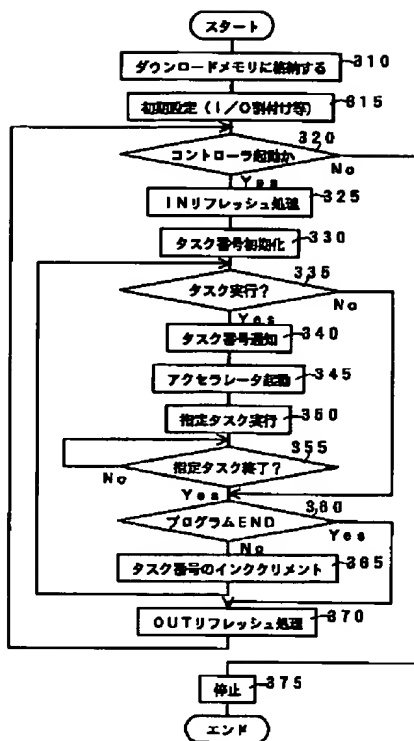
【図15】



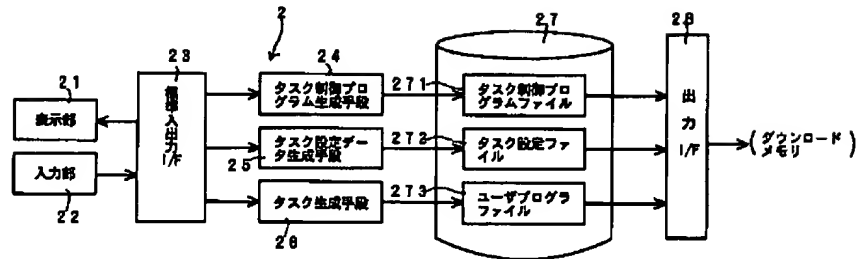
【図12】



【図13】



【図14】



フロントページの続き

Fターム(参考) 5H215 BB10 CC05 CX02 GG04 GG09
 GG20 HH03
 5H219 CC10 HH25 HH28
 5H220 BB12 CC05 CX02 DD01 DD10
 EE03 EE08 FF05 JJ12 JJ19
 JJ26 JJ29 JJ59
 9A001 BB01 DD01